



# **KAASUHÄLYTYSJÄRJESTELMÄN PÄIVITYS**

Sami Puntti

Opinnäytetyö  
Kesäkuu 2014  
Tietotekniikka  
Ohjelmistotekniikka

## TIIVISTELMÄ

Tampereen ammattikorkeakoulu  
Tietotekniikan koulutusohjelma  
Ohjelmistotekniikan suuntautumisvaihtoehto

## PUNTTI SAMI

Kaasuhälytysjärjestelmän päivitys

Opinnäytetyö 28 sivua

Kesäkuu 2014

---

Tässä työssä käsitellään yksityisen sairaalan kaasuhälytysjärjestelmän hälytystavan laajennusta. Tällä hetkellä järjestelmä lähettää tekstiviestin yhteen matkapuhelimeen. Viestissä on vikaa kuvaava numero ja pidempi tekstimuotoinen selitys viasta. Numeron perusteella pystytään päättämään nopeasti hälytyksen vakavuus. Pidemmästä tekstistä selviää tarkemmin, mistä hälytyksessä on kysymys. Hälytys tulee, kun kaasua on tilattava lisää, kun järjestelmään tulee häiriö ja paine laskee liian pieneksi tai nousee liian suureksi. Hälytys tulee myös, jos jossain kompressorissa tai kaasukeskuksessa ilmenee vikaa.

Tarkoituksena on muuttaa järjestelmää niin, että palvelin lähettää ilmoituksen Integralin palvelimelle. Kyseessä on Linux-pohjainen palvelinratkaisu, johon asennetaan Kannel ohjelma. Kannelia käytetään tekstiviestien hallintaan. Palvelimeen liitetään SMS-modeemi, jonka avulla pystytään vastaanottamaan tekstiviestejä ja lähettämään niitä useisiin puhelimiin. Lisäksi tehdään käyttöliittymä HTML- ja PHP-ohjelmointikielillä, jonka avulla monen matkapuhelinnumeron ja sähköpostiosoitteen hallinta on helppoa.

Käyttöliittymän yhtenä tavoitteena on pystyä lisäämään ja poistamaan matkapuhelinnumeroita ja sähköpostiosoitteita. Toisena tavoitteena on pystyä määrittämään eri tärkeysasteisille viesteille eri vastaanottajat. Esimerkiksi kaasun tilausviestit ja muut kiireettömät viestit voidaan toimittaa ainoastaan Integralin asentajien yleiseen sähköpostiosoitteeseen, josta palvelupyynnot käsitellään seuraavan työpäivän aikana. Toisaalta heti toimintaa vaativat hälytykset pystytään ohjaamaan useaan matkapuhelimeen ja sähköpostiosoitteeseen, jotta asia tulee varmasti hoidettua, eikä vastuussa ole ainoastaan yksi henkilö.

---

Asiasanat: hälytysjärjestelmät, kaasu, käyttöliittymä, tekstiviesti.

## ABSTRACT

Tampereen Ammattikorkeakoulu  
Tampere University of Applied Science  
Degree Programme in ICT Engineering  
Option of Software engineering

SAMI PUNTTI

An upgrade of a gas alert system

Bachelor's thesis 28 pages

June 2014

---

This thesis deals with the expansion of the alarm methods in the gas alert system in a private hospital. At the moment the system sends an sms to one mobile phone. The message contains an error code and an explanation of the error. The urgency of the alert can be quickly assessed based on the error code. The explanation offers more detailed information on the alert. An alarm is triggered when more gas needs to be ordered, when there is a problem with the system and pressure falls too low or rises too high, or when there is a malfunction in one of the compressors or the gas central.

The aim is to modify the system to send a notification to an Integral server, which is a Linux based server solution installed with the Kannel program. Kannel is used in sms management. An sms modem is attached to the server, which allows the system to receive messages and send them to multiple phones. HTML and PHP programming are used to create an interface for easy management of multiple phone numbers and e-mail addresses.

One of the goals of the interface is to be able to add and remove phone numbers and e-mail addresses. The other goal is to be able to specify different recipients for error alerts depending on the urgency of the alert. Non-urgent messages, e.g. gas refill orders can be sent only to the general email address of Integral maintenance, from where the service request will be handled during the next working day. Alerts requiring immediate action can be sent to multiple mobile phones and e-mail addresses to ensure a response and to make sure no one person is responsible.

---

Key words: alert system, gas, user interface, text message

## SISÄLLYS

1	Johdanto .....	6
2	Työn esittely .....	8
2.1	Työn tilaaja .....	8
2.2	Vanha järjestelmä .....	8
2.3	Hyvät puolet .....	9
2.4	Huonot puolet .....	10
3	Tekniikat ja työkalut .....	11
3.1	Kannel .....	11
3.2	Käytetyt ohjelmointikielet .....	11
4	Järjestelmän määrittely .....	13
4.1	Hälytysjärjestelmän määrittely .....	14
4.2	Hallintasivun käyttöliittymän suunnittelu .....	14
4.3	Hälytysviestien tulkinta / parsiminen .....	16
4.4	Käyttöliittymä .....	16
4.5	Hyvät puolet .....	17
4.6	Huonot puolet .....	18
5	Toteutus ja testaus .....	19
5.1	Kannelin konfigurointi .....	19
5.2	Tekstiviestin lähetys .....	21
5.3	Hylätyt ratkaisut .....	22
5.4	Testaus .....	22
5.5	Virhetilanteet .....	24
5.6	Käyttöönotto .....	25
6	Yhteenveto .....	26

## LÄHTEET

## LYHENTEET JA TERMIT

SMS	Short Message Service, puhelimissa käytetty tekstiviesti
WAP	Wireless Application Protocol, kokoelma kielistä ja työkaluista, sekä alusta internet tyyppisten palveluiden tuomiseksi puhelimiin.
PHP	Lyhenne sanoista PHP: Hypertext Preprocessor. Erityisesti dynaamisten Web-sivujen luontiin tarkoitettu ohjelmointikieli.
HTTP	HyperText Transfer Protocol, käytetään internetissä määrittämään miten viestit muodostetaan ja siirretään.
ICT	Information and communication technology, eli tieto- ja viestintäteknologia (TVT). Tarkoitetaan kaikkea tekniikkaa, mikä liittyy tietojen käsittelyyn.
SIM	Subscriber Identity Module, eli tavallinen matkapuhelinliittymän älykortti, joka laitetaan kännykän sisään.
PIN	4 numeroinen koodi, jolla SIM-kortin saa auki. Oletuksena ”0000” tai ”1234”.

## 1 JOHDANTO

Tämän työn tarkoituksena on kehittää Integral Oy:n hallinnoimaa yksityisen sairaalan kaasuhälytysjärjestelmää eteenpäin käyttäjäystävällisemmäksi. Kyseinen järjestelmä on ollut käytössä jo kauan ja ennen kaikkea järjestelmä on ollut käytössä jo, ennen kuin sairaala on ollut Integral Oy:n asiakkaana. Vanhassa järjestelmässä hälytyksen sattuessa lähetetään tekstiviesti yhteen matkapuhelimeen. Tästä seuraa järjestelmän kriittisin ongelma, eli yksi henkilö oli vastuussa hälytykseen reagoimisesta. Kappaleessa kaksi käydään tarkemmin läpi työn tilaajayritystä, itse työtä, sekä vanhaa järjestelmää, jota tällä työllä on tarkoitus kehittää.

Työ toteutetaan pääasiassa PHP-ohjelmointikielellä. Lisäksi toteutetaan hallintapaneeli HTML-, sekä PHP-ohjelmointikielillä, jolla hallitaan tulevista hälytyksistä vastuussa olevia henkilöitä. PHP-ohjelmointikieli oli luonnollinen valinta työn toteuttamiseen, koska se on Integralin eniten käyttämä ohjelmointikieli. Asiantuntemusta ja apua valituista tekniikoista löytyy läheltä ja tarvittaessa järjestelmän jatkokehitykseen löytyy paljon osaamista ja ideoita. Kaikki PHP:n käyttöön vaaditut palvelimet ja ympäristöt ovat jo valmiina toiminnassa, eikä ylimääräisiä testejä ja suunnitelmia tämän osalta tarvita. PHP sopii hyvin verkkopohjaisen hallintapaneelin toteuttamiseen, koska se ei vaadi selaimelta erillistä tukea. Myös tiedostojen ja vaadittujen tietokantojen käyttö palvelimelta onnistuu helposti.

Edellä mainittu järjestelmän tekstiviesti on tarkoitus vastaanottaa ja jatkolähetää Kannel-nimisellä ohjelmalla. Kannel on yksi suosituimmista tarjolla olevista avoimen lähdekoodin SMS-yhdyskäytävistä. Yksi Kannelin suurimpia vahvuuksia on toimintavarmuus suurillakin tekstiviestimäärillä (Drupal, Kannel projekti). Kyseessä on erittäin valmis paketti tekstiviestien käsittelyyn. Yksi Kannelin käytön suurimmista haasteista on tarvittavan SMS-modeemin ja Kannel-ohjelman toiminnan yhdistäminen. Tämä hoidetaan pääasiassa Kannelin konfigurointitiedostolla. Kannel-projektin aloitti Wapit Ltd ja nykyään kehityksestä ja lisenssistä vastaa the Kannel Group. Lisää käytetyistä tekniikoista ja Kannelista käsitellään kappaleessa kolme. (Kannel Group)

Työn tarkoituksena on kehittää kaasuhälytysjärjestelmää niin, että ilmoitus häiriöistä on mahdollista ohjata moneen matkapuhelimeen, sekä haluttuihin sähköpostiosoitteisiin.

Lisäksi toteutetaan viestejä tulkitseva toiminnallisuus, jolla eritasoiset viestit voidaan ohjata eteenpäin eri laajuudella. Eritasoisten viestien vastaanottajia on tarkoitus pystyä hallitsemaan helposti verkkopohjaisella hallintapaneelilla. Kappaleessa neljä käsitellään lisää toteutettavaa järjestelmää, sen vaatimuksia sekä hallintapaneelin käyttöliittymää.

Kappaleessa viisi käsitellään työn toteutusta ja testausta. Sekä lopuksi kappale kuusi pitää sisällään tämän työn yhteenvedon.

## 2 TYÖN ESITTELY

Tässä kappaleessa käydään läpi työn tilaajaa, vanhaa järjestelmää ja sen toimintaa.

Vanha järjestelmä on ollut toiminnassa jo vuosia, eikä siitä muuteta mitään. Se jää hyväksi pohjaksi ja muutokset tehdään jatkoksi tekstiviestille, jonka vanha järjestelmä lähettää. Tästä luonnollisesti jää järjestelmään heikko kohta ja on mahdollista, että operaattorin vian vuoksi koko tekstiviesti-ilmoitus jää lähtemättä. Tälle ei valitettavasti tällä hetkellä voida tehdä mitään, eikä sitä käsitellä enempää tässä työssä.

### 2.1 Työn tilaaja

Alkuperäinen työn tilaaja oli European IT Oy, keskikokoinen ICT-alan yritys, joka oli juuri saanut palveluoperaattorin statuksen. Yritys työllisti noin kaksikymmentä työntekijää. European IT tarjosi asiakkaille lähes kaikki heidän tarvitsemansa ICT-palvelut. Esimerkkeinä Hosted Exchange sähköpostijärjestelmä, Voip-puhejärjestelmät, matkapuhelinliittymät, varmuuskopiointi, asiakkaan tarpeisiin kustomoitava CRM-asiakashallinta järjestelmä ja IT-laitteet puhelimista palvelimiin esiasennettuina. Itse toimin lopputyöni ohella yrityksessä IT-tukihenkilönä. Syksyllä 2013 European IT yhdistyi Integral Oy:n kanssa yhdeksi suureksi yritykseksi.

Integral Oy on yli 40 ICT-ammattilaista työllistävä yritys, jolla asiakkaina on kaiken kokoiset yrityksiä. Yritys on toiminut vuodesta 2005 ja liikevaihto on noin 4,5 miljonna euroa. Integral tarjoaa myös asiakasyrityksilleen kaiken tietotekniikan, mitä yritys tarvitsee. Palveluina esimerkiksi pilvipalveluja, tietoturvaa, lähiverkkoa, palvelunhallintaa, hostingia, käyttäjätukea, etävarmistusta, laitteita, sovelluksia, räätälöityjä järjestelmiä, konsultointia ja luonnollisesti edellä mainitut European IT:n tarjoamat palvelut. (Integral)

### 2.2 Vanha järjestelmä

Alkuperäinen kaasuhälytysjärjestelmä pystyy lähettämään yhden tekstiviestin. Puhelinta, johon tekstiviesti tulee, joutuu jatkuvasti päivystämään yksi henkilö. Viestejä on eri tasoisia ja numerosta viestin edessä näkee, minkä tason viesti on kyseessä. Yleisimpiä



viestejä ovat sairaalakaasujen lisätilauskehotteet, jotka yleensä eivät edellytä toimenpiteitä, koska kaasua tilataan automaattisesti tiettyinä päivinä. Tilausviestin tullessa automaattitilausta edeltävänä päivänä, viesti ei aiheuta toimenpiteitä.

Viestit, jotka on merkitty hälytystä vastaavalla numerolla, vaativat välitöntä toimintaa ja tilanteen tarkastusta. Päivystäjän on välittömästi otettava yhteys sairaalaan ja tarkistettava, ovatko paikallaolijat jo havainneet hälytyksen ja mahdollisesti toimineet tilanteen vaatimalla tavalla. Jos tilannetta ei ole hoidettu tai paikalla ei ole ketään, joka voi sen hoitaa, on päivystäjän lähdettävä itse paikalle hoitamaan asia.

Hälytyksen voi siis havaita myös sairaalan kellarin kaasuhuoneen ulkopuolella vilkkuvasta hälytysvalosta. Punainen valo ilmoittaa, että happikeskuksessa on vika ja mahdollisesti vaarallisen korkea O<sub>2</sub>-pitoisuus. Punaisen valon palaessa on ennen huoneeseen menoa aina tarkistettava kaasumittareista O<sub>2</sub>-pitoisuus. Jos pitoisuus on liian korkea, huoneen ovea ei saa avata ilman huoltomiestä. Huoneen ulkopuolella on myös kyltti, joka ilmoittaa, onko vika havaittu ja hoidettu tai huolto tilattu, vai onko kyseessä ensimmäinen paikalle tullut henkilö. Kellari kuulostaa paikalta, jossa kukaan ei kävisi, mutta tässä tapauksessa tämä kyseinen käytävä on sisäkautta ainoa tie sairaalan ja hallinnon erillisten siipien välillä, joten ohi kävellään usein. Kellarissa on myös röntgenkuvauksen vaatimia hyvin eristettyjä tiloja.

### 2.3 Hyvät puolet

Vanhan järjestelmän paras puoli on sisältyvien laitteiden vähyys. Järjestelmässä on vikoja tulkitseva toiminnallisuus, joka antaa hälytyksen, tekstiviestin lähetystoiminto, sekä puhelin, joka ottaa viestin vastaan. Tässä järjestelmässä ei ole juuri esiintynyt hälytysjärjestelmän laitteista johtuvia vikoja.

Viestit tulevat vain yhdelle henkilölle, joten epäselvyyttä vastuullisesta henkilöstä ei voi syntyä.

## 2.4 Huonot puolet

Viestin kohdentamisessa yhdelle henkilölle esiintyy myös heikkous, sillä järjestelmä on erittäin vaativa päivystävälle henkilölle. Puhelin täytyy olla kokoajan mukana ja päivystäjän täytyy olla jatkuvasti työkykyinen.

Valo kaasukeskuksen ulkopuolella, joka ilmoittaa viasta, ei ole erityisen näkyvällä paikalla. Hälyytysvalo näkyy vain käytävällä, jossa kaasukeskus on. Jos kukaan ei satu kävelemään ohi, se voi jäädä huomaamatta.

### 3 TEKNIIKAT JA TYÖKALUT

Tässä kappaleessa esitellään lyhyesti Kannelia ja kerrotaan sen ominaisuuksista ja historiasta. Kannelin käyttämiseen soveltuu parhaiten Linux-käyttöjärjestelmä. Kannelista on tehty myös windows-käyttöjärjestelmässä toimiva versio, mutta se ei luonnollisesti ole yhtä suosittu kehittäjien keskuudessa, kuin Linux-versio. Lisäksi käydään lyhyesti läpi Kannelin hallintaan valittua ohjelmointikieltä PHP:ta.

#### 3.1 Kannel

Kannel on avoimen lähdekoodin WAP ja SMS yhdyskäytävä. Kannel-projekti aloitettiin vuonna 1999 WapIT Ltd:n toimesta. Nykyään Kannel-projektin kehityksestä vastaa the Kannel Group. Kannel tarjoaa kaikille mahdollisuuden käyttää ja tarjota eteenpäin WAP-palveluita tarjoamalla suuren osan WAP-infrastruktuurista kaikkien saataville avoimella lähdekoodilla. Kannel on lisensoitu the Kannel Groupin omalla Kannel Software-lisenssillä, joka on samankaltainen, kuin Apache-lisenssi. (Kannel käyttöohje, 1.)

Kannel on myös tarkoitettu mahdollistamaan HTTP-pohjaisten palveluiden yhdistämisen puhelinten SMS-ominaisuuksiin. Tämän vuoksi Kannel sopii erinomaisesti käytettäväksi tässä projektissa. Kannelia on kutsuttu maailman nopeimmaksi SMS-yhdyskäytäväksi, tämä luonnollisesti on hyvä puoli projektia ja sen toimintaa ajatellen. (Kannel käyttöohje, 1.)

#### 3.2 Käytetyt ohjelmointikielet

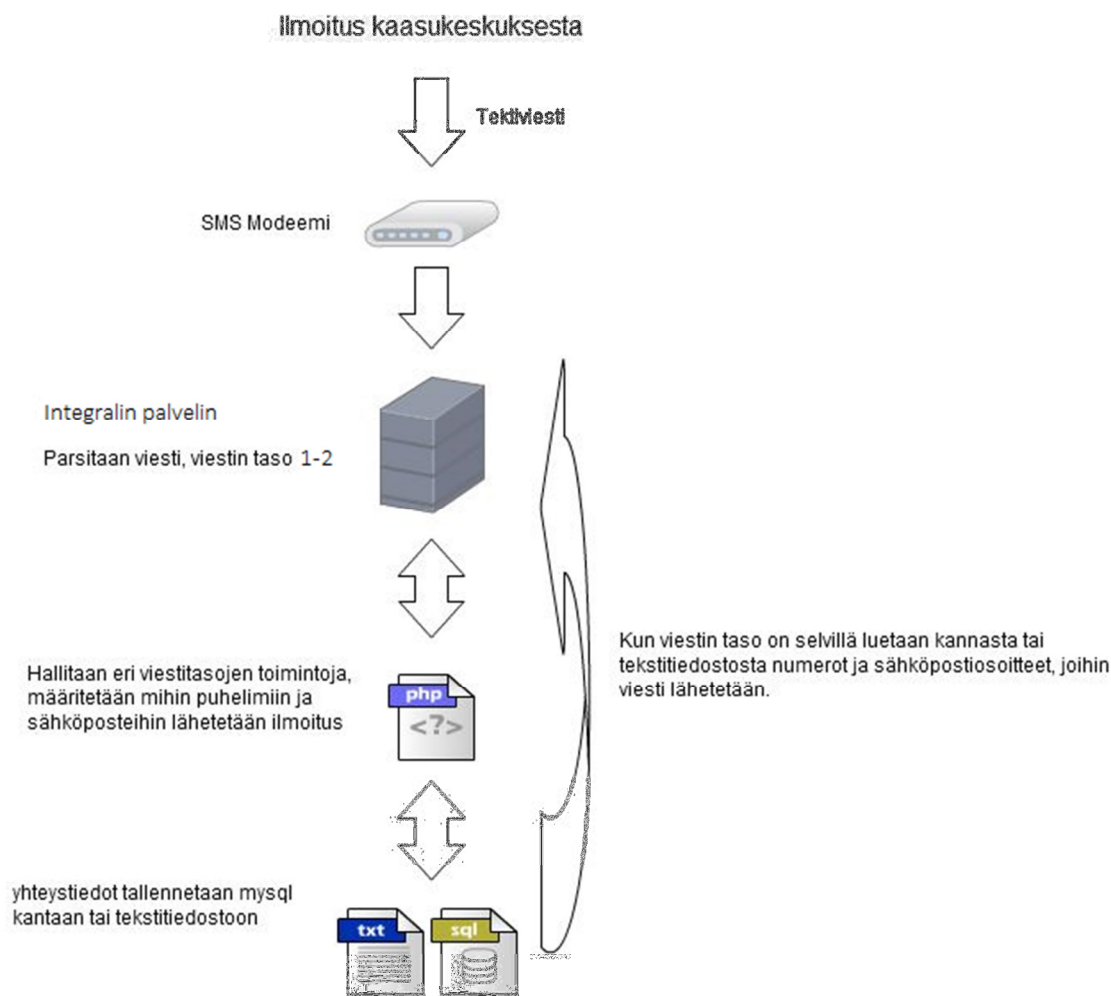
PHP on erityisesti dynaamisten web-palvelujen luomiseen tarkoitettu ohjelmointikieli. PHP:n käyttö vaatii erillisen palvelimen, joka sisältää PHP-tulkin. Web-sivun latauksessa selain ei käsittele ollenkaan PHP-koodia, vaan palvelimen PHP-tulkki kääntää koodin selaimen ymmärtämään muotoon ja tästä syystä selaimelta ei vaadita erillistä tukea PHP:ta varten. (W3schools.com; The PHP Group)

Ohjelmointikielen valinta perustui erityisesti tilaajayrityksen tarpeisiin ja heidän vahvuuksiinsa. Integralin tarjoamista järjestelmistä, sekä heidän omista muokattavista työ-

kaluistaan suuri osa on toteutettu PHP:lla, joten yrityksestä löytyy vahvaa osaamista tästä ohjelmointikielestä. PHP:n sisäistäminen on nopeaa, käyttäminen miellyttävää ja testiympäristön luominen onnistuu vaivattomasti. PHP on erinomainen valinta ohjelmointikieleksi Kannelia hallitsevalle web-sovellukselle. (W3schools.com; The PHP Group)

#### 4 JÄRJESTELMÄN MÄÄRITTELY

Tässä kappaleessa käydään läpi järjestelmän toimintaa ja toimintaperiaatteita. Järjestelmä itsessään on melko yksinkertainen. Vastaanotetaan tekstiviesti, tutkitaan mitä viestissä lukee ja lähetetään viesti eteenpäin. Järjestelmän toiminnan ymmärtämistä helpottaa kuva 1.



Kuva 1. Suunniteltu kuvaus järjestelmän toiminnasta

Kuten kuvasta 1 selviää, kaasujärjestelmä lähettää tavallisen tekstiviestin. SMS-modeemi vastaanottaa viestin ja Kannel ohjelma, joka hallitsee SMS-modeemia, ohjaa viestin eteenpäin Integralin palvelimelle kehitettyyn PHP-sovellukseen. Sovellus tulkitsee viestin ja ohjaa jatkotoimia hälytyksen tasosta riippuen. Viesti lähetetään eteenpäin tekstiviestillä tai sähköpostilla tarpeen mukaan.

#### 4.1 Hälytysjärjestelmän määrittely

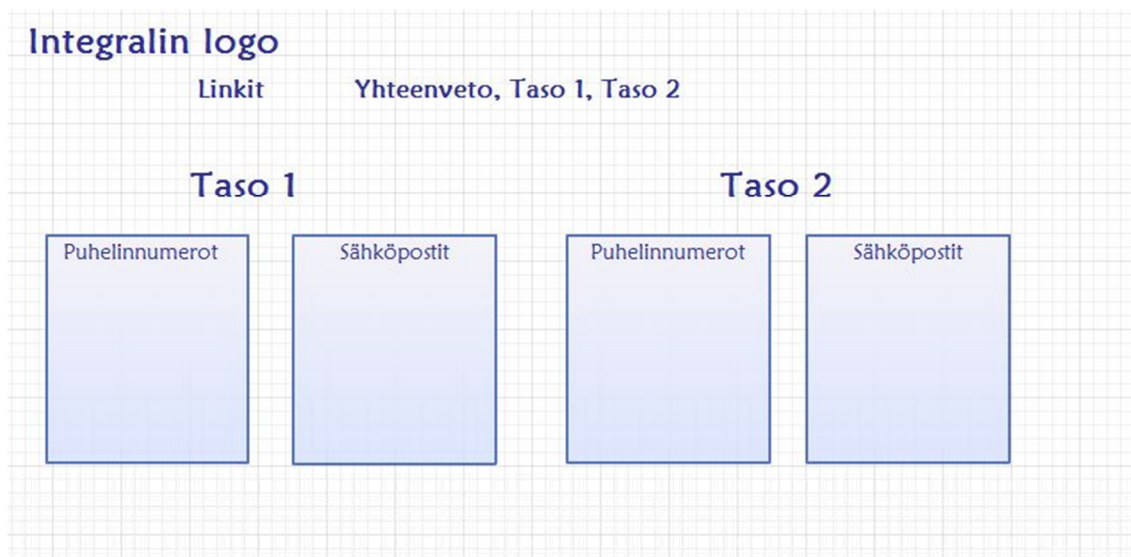
Järjestelmän määrittelyt tehtiin yhdessä Integralin kehitysjohtaja Cristoffer Eleniuksen kanssa. Seuraavassa pääkohdat määrittelystä:

- Vanha kaasujärjestelmä lähettää tekstiviestin, tätä osaa järjestelmästä ei muuteta. Järjestelmään tehdään lisäominaisuus.
- Viesti vastaanotetaan Sierra Wireless -merkkisellä SMS modeemilla.
- Hälytys Integralin palvelimelle, testausvaiheessa tässä käytetään paikallista tietokonetta, johon on asennettu Linux-käyttöjärjestelmä.
- Viesti parsitaan ja tutkitaan, miten on toimittava
  - Hälytys: viesti lähtee useaan puhelimeen ja sähköpostiin, vastuuhenkilöiden hallinta tapahtuu hallintapaneelissa
  - Kaasun lisätilaus tai vastaava ei kiireellinen tapaus: viesti sähköpostilla yleiseen sähköpostiosoitteeseen, hallinta tapahtuu hallintapaneelissa
- Lähetetään viestit Kannelin hallinnoimalla SMS-modeemilla määritettyihin puhelinnumeroihin ja sähköpostit Integralin sähköpostipalvelinta käyttäen määritettyihin sähköposteihin.

#### 4.2 Hallintasivun käyttöliittymän suunnittelu

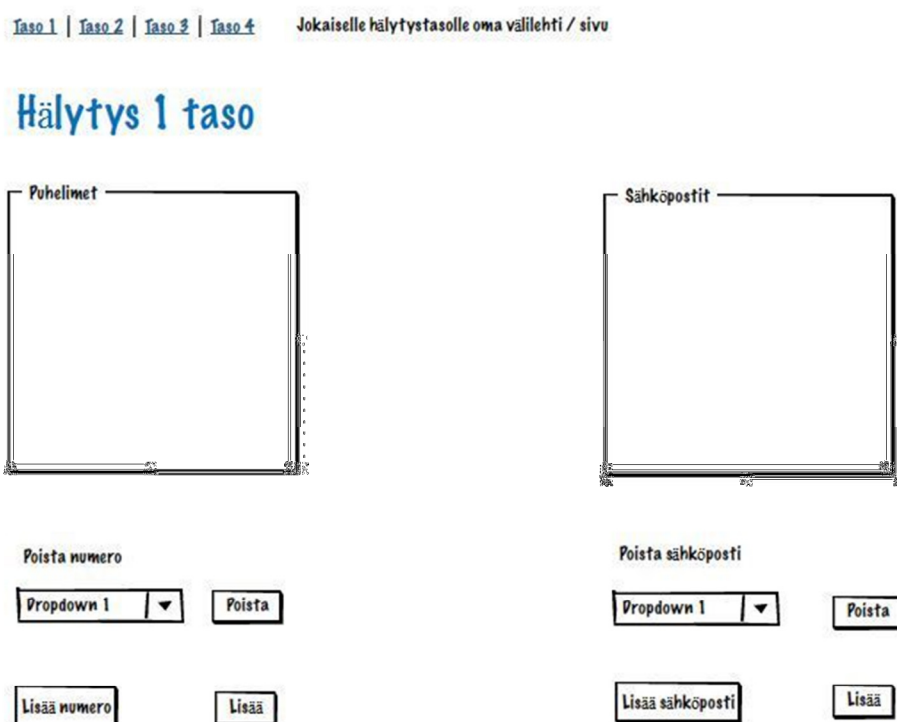
Hallintasivusto on internetsivusto, johon on tarkoitus päästä helposti hallitsemaan henkilöitä, jotka ovat vastuussa hälytysviestien hoitamisesta. Vastuuhenkilö tulee saamaan viestin sähköpostilla ja tekstiviestillä tai tarvittaessa toinen vaihtoehto voidaan jättää pois.

- Toteutus HTML ja PHP-ohjelmointikielellä
- Toteutetaan tätä tarkoitusta varten oma hallintasivusto
- Ominaisuudet:
  - Vastaanotetun viestin eri tasojen vastuuhenkilöiden hallinta: pystyttävä helposti vaihtamaan puhelinnumeroita ja sähköpostiosoitteita, johon tietyllä hälytystasolla tulevat viestit lähetetään.
  - Selkeä yhteenveto, kuten kuvassa 2, josta näkee nopeasti tarvitaanko hälytystasoihin muutoksia.



Kuva 2. Hallintasivuston yhteenvetosivu

Hallinnan yhteenvedossa on eritelty sähköpostit ja puhelinnumerot, joihin viesti lähetetään. Linkeistä Integralin logon ja tasojen 1 ja 2 otsikoiden välissä päästään hallinnoimaan eritasoisten viestien lähetyslaajuutta. Seuraava kuva 3 selvittää lisää numeroiden ja sähköpostiosoitteiden lisäyksestä ja poistamisesta.



Kuva 3. Hahmotelma hallintasivuston numeroiden ja sähköpostien hallinnasta

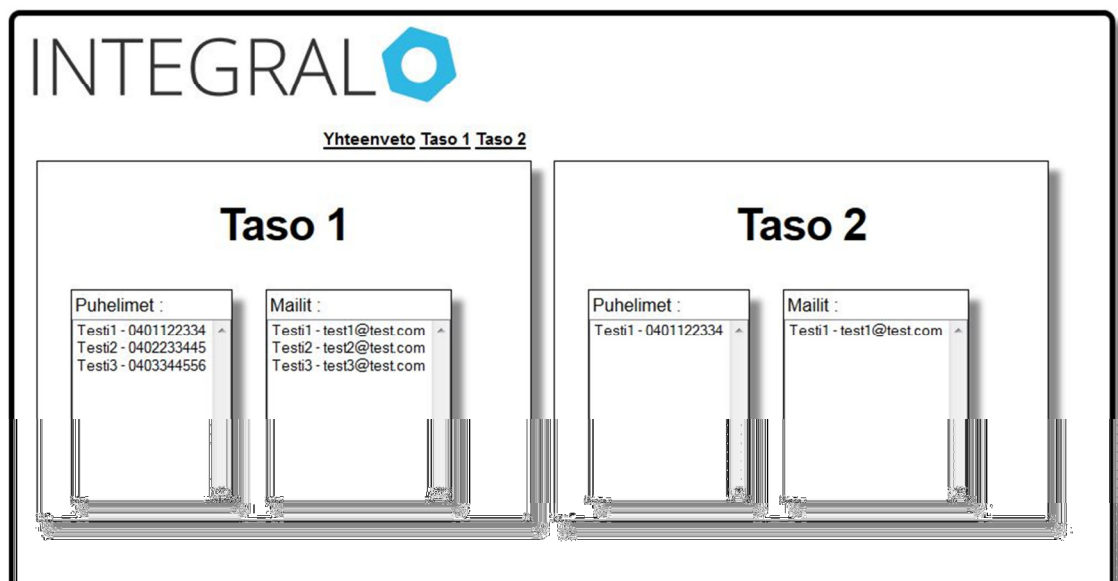
Puhelinnumeroiden ja sähköpostien hallinta on hyvin yksinkertaista. Kuvan 3 listassa näkyy tämän hetkiset vastuuhenkilöt kyseisellä tasolla. Henkilöiden poistamisen ja lisäämisen kentät ja tarvittavat napit on sijoitettu sivun alareunaan.

### 4.3 Hälytysviestien tulkinta / parsiminen

Hälytysviestit erotellaan viestin sisältämän tekstin perusteella. Viestejä on periaatteessa kahdella tasolla. Tasolla 1, eli kiireellisesti hoidettava = ”happivuoto” ja tasolla 2, eli kiireetön hälytys. Tasolla 2 hälytyksiä on monta erilaista: happimittausvika, happitilaus, happikeskusvika, ilokaasutilaus, ilokaasumittausvika, ilmatilaus, ilokaasukeskusvika, kompressorin 1- vika ja kompressorin 2- vika. 2-tason hälytyksistä ilmoitetaan joko sähköpostilla tai puhelimitse eteenpäin asiasta vastaavalle taholle hälytyksestä riippuen.

### 4.4 Käyttöliittymä

Ulkoasusta tehtiin pelkistetty ja vaalea. Kuvissa 4 ja 5 näkyy lopullinen yhteenveto- ja taso 1- sivujen ulkoasu. Käyttöliittymä on yksinkertainen, eikä mitään tarpeetonta ole näkyvissä. Käyttäjä pystyy ainoastaan lisäämään ja poistamaan käyttäjiä.



Kuva 4. Lopullinen yhteenvedon ulkoasu



INTEGRAL

[Yhteenveto](#) [Taso 1](#) [Taso 2](#)

## Taso 1

Puhelimet :

- Testi1 - 0401122334
- Testi2 - 0402233445
- Testi3 - 0403344556

Poista numero :  
 --- Select ---

Uusi numero :  
 Nimi :   
 Numero :

Mailit :

- Testi1 - test1@test.com
- Testi2 - test2@test.com
- Testi3 - test3@test.com

Poista maili :  
 --- Select ---

Uusi maili :  
 Nimi :   
 Maili :

Kuva 5. Lopullinen taso 1 sivun ulkoasu

Poistettava henkilö valitaan Kuvan 5 alasvetovalikosta ja poistaminen kuitataan poistapainilla. Henkilön lisääminen vastaavasti tapahtuu kirjoittamalla henkilön nimi ensimmäiseen kenttään ja numero tai sähköpostiosoite toiseen kohtaan ja lopuksi toiminto kuitataan lisää-painilla.

Järjestelmä on suunniteltu yrityksen sisäiseen käyttöön, joten ei ole tarpeellista toteuttaa tarkastuksia puhelinnumeroiden ja sähköpostiosoitteiden oikeinkirjoituksen suhteen. Oletetaan, että järjestelmän käyttäjä tarkastaa oikeinkirjoituksen itse.

#### 4.5 Hyvät puolet

Kuten edellä jo mainittiin, yksi merkittävimmistä parannuksista järjestelmässä on, että hälytys ei ole enää yhden henkilön varassa. Todennäköisyys asian hoitamattomuuteen pienenee huomattavasti ja käyttömukavuus lisääntyy merkittävästi. Kun hälytykset tulevat usealle henkilölle, joku on varmasti hereillä ja pystyy ottamaan asian hoitoonsa.

Kaasun lisätilaukset ovat tähän asti tulleet samanarvoisena tekstiviestinä, kuin kriittiset hälytykset ja niihin on täytynyt reagoida vaikka keskellä yötä. Nyt uudessa järjestelmässä saadaan tilaukset ja muut kiireettömät viestit eriteltyä hälytyksistä ja ohjattua ne esi-

merkiksi tiettyyn sähköpostiosoitteeseen odottamaan vapaata henkilöä hoitamaan asiaa ja näin saadaan päivystäjältä turhat hälytykset pois.

#### 4.6 Huonot puolet

Todennäköisyys siihen, että jokainen jolle kriittisen hälytyksen viesti tulee, olisi työkyvytön on hyvin pieni, mutta tilanne on mahdollinen. Tästä syystä päivitettyssäkin järjestelmässä pitäisi jollain tavalla vielä varmistaa ja sopia, että joku on aina päivystämässä puhelintaan tai sähköpostiaan tai vastaavasti voidaan sopia, että ilmoitetaan muille päivystäjille, ettei esimerkiksi aiota olla vastuullisia tiettynä aikana. Näin muut päivystäjät tietävät olla entistä tarkempina asian suhteen.

Toinen ongelma on estää asioiden moneen kertaan hoitaminen. Miten ilmoitetaan muille, että asia on hoidettu? Tämä voidaan esimerkiksi ratkaista sähköpostilistalla, jolloin ongelman hoitaja laittaa tietylle ryhmälle viestin ongelman ratkaisemisesta. Toinen mahdollinen ratkaisu tähän olisi laajentaa hallintasivustoa niin, että muut käyttäjät voisivat tarkistaa ennen toimimista, onko asia jo hoidettu.

## 5 TOTEUTUS JA TESTAUS

Seuraavaksi käsitellään järjestelmän toteutuksen olennaiset kohdat ja eri osa-alueet. Ensiksi käsitellään Kannelin konfigurointitiedostoa ja sen avulla tekstiviestin vastaanottoa. Seuraavaksi tekstiviestin lähetystä ja lopuksi käsitellään järjestelmän testausta ja käyttöönottoa.

### 5.1 Kannelin konfigurointi

The Kannel Groupin internetsivuilla on kattava käyttöohje järjestelmän käyttäjille. Käyttöohjeesta löytyy selitys jokaisesta konfigurointitiedoston määritteestä. Sivuilta löytyy myös usein kysytyt kysymykset ja paljon valmiiksi ratkaistuja ongelmatapauksia, joista löytyy mahdollisia vihjeitä myös oman järjestelmän ongelmiin.

Kannelin konfigurointitiedosto on eritelty ryhmiin, jotka määrittävät eri osia järjestelmän toiminnasta. Jokainen ryhmä määritellään erikseen, esimerkiksi ”*group = core*” komennolla, joka on aina ryhmän ensimmäisellä rivillä. Ryhmissä ei voi olla tyhjiä rivejä välissä, ryhmien välillä tyhjiä rivejä saa olla. Core-ryhmässä määritellään Kannelin perusominaisuudet, kuten pääkäyttäjän salasana ja ip-osoite, josta järjestelmää voidaan hallita. (Kannel käyttöohje, 3)

*admin-password = xxxx // pääkäyttäjän salasana selainpohjaista käyttöä varten.*

*status-password = xxxx // salasana kannelin tilan tarkastukseen selaimella.*

*admin-deny-ip = "\*.\*.\*.\*" // estetään käyttö kaikista ip-osoitteista.*

*admin-allow-ip = "127.0.0.1" // sallitaan käyttö vain paikallisesti.*

Muita ryhmiä ovat: smsc, smsbox, send-smsuser, modems ja sms-service. Smsc-ryhmässä määritetään kaikki tiedot, mitä tarvitaan viestikeskukseen yhdistämiseen. Esimerkiksi SMS-modeemin käyttämä USB-portti, lähettäjän numero, SIM-kortin PIN-koodi, sekä käytettävän viestikeskuksen numero. (Kannel käyttöohje)

*device = /dev/ttyUSB0 // Kannel määritetään etsimään SMS-modeemia palvelimen USB-portista 0.*

*pin = 0000 // määritetään SIM-kortin käyttämä PIN-koodi.*

*my-number = +35845xxxxxx // määritetään oman SIM-kortin numero.*

*sms-center = +358447983500 // määritetään käytettävä viestikeskus, johon tekstiviestit lähetetään.*

Smsbox-ryhmässä määritetään esimerkiksi tietokoneen portti, josta tekstiviestin lähetystä hallitaan, sekä oma numero, joka liitetään viesteihin.

*sendsms-port = 13014 // portti, jonka kautta webhallinta lähettää viestejä.*

*global-sender = "+35845xxxxxxx" // oma numero, joka liitetään lähetettyihin viesteihin.*

Sms-services-ryhmästä löytyy yksi olennaisimmista määrittäyksistä. Tässä ryhmässä määritetään kohde, johon vastaanotettu tekstiviesti ohjataan.

*get-url = http://127.0.0.1/xxx/sms\_receive.php?phone=%p&text=%a*

Tässä määritetään, mistä vastaanottava koodi löytyy, eli paikalliselta koneelta, tietystä kansista, tiedostosta nimeltä sms\_receive.php. Parametreina kyseiselle tiedostolle annetaan lähettäjän puhelinnumero ja koko viestin sisältämä teksti. Tähän määrittäykseen voidaan lisätä hyvin paljon erilaista tietoa viestistä. Viestin mahdollisia lisämäärittäjiä ovat esimerkiksi vastaanottoaika, ensimmäinen sana tekstistä, toinen sana tekstistä, loppu teksti, vastaanottajan numero, viestissä käytetty merkitseminen, viestin ID-tieto ja paljon muuta. Muita tämän ryhmän mahdollisia määrittäjiä ovat sallitut ja estetyt puhelinnumeron alku numerot, sekä paljon muita vaihtoehtoja, joilla voidaan määrittää ja rajoittaa vastaanotettavia viestejä. (Kannel käyttöohje, 7)

Tekstiviestin lähetykseen vaaditaan myös vähintään yksi määritetty sendsms-user-ryhmä, jossa määritetään kannelin käyttäjä. Näillä tiedoilla koodista voidaan pyytää Kannelia lähettämään tekstiviesti. Määrittäjiä ovat esimerkiksi käyttäjän nimi ja salasana, ip-osoitteet josta käyttö sallitaan tai estetään ja pitkän viestin mahdollinen pilkkominen lyhyempiin osiin lähetyksen mahdollistamiseksi.

*username = xxxx // käyttäjän nimi.*

*password = xxxx // käyttäjän salasana.*

*user-deny-ip = "\*.\*.\*.\*" // estetään käyttö kaikista ip-osoitteista.*

*user-allow-ip = localhost // sallitaan käyttö vain paikallisesti.*

*max-messages = 2 // jos lähetetyn viestin merkkien määrä ylittää sallitun tekstiviestin merkkien määrän, tämä määrittys mahdollistaa viestin pilkkomisen määrityksen mukaisesti kahteen erilliseen tekstiviestiin.*

Kannelin konfigurointitiedostossa on halutuista määrityksistä riippuen jopa sata riviä tekstiä ja jokainen väärä rivi voi aiheuttaa järjestelmän toimimattomuuden tai altistaa järjestelmän ulkoisille vaaratekijöille. Kannelissa on kaikille asetuksille myös oletusarvo, jos riviä ei kirjoiteta konfigurointitiedostoon, järjestelmä käyttää automaattisesti määrityksen oletusarvoa. (Kannel käyttöohje, 7)

## 5.2 Tekstiviestin lähetys

Kannelille tekstiviestin lähetyskomento annetaan internet selaimella. Osoiteriville kirjoitetaan Kannelin ymmärtämä lähiverkon osoite ja parametreiksi kaikki viestin lähetykseen vaaditut tiedot. Seuraavassa esimerkki oman Kannel-konfiguraationi tekstiviestin lähetyskomennosta.

*http://127.0.0.1:13014/cgi-bin/sendsms?username=xxxx&password=xxxx&smc=gas-it&from=1234&to=0407765392&text=Testausta*

Samoin kuin tekstiviestin vastaanotossa, Kannelille täytyy ensin kertoa, mitä ohjelman pitää tehdä. Kutsutaan Kannelin omaa sendsms-toimintoa paikalliselta tietokoneelta. Komennossa kaksoispisteen jälkeinen numerosarja on Kannelille määritetty portti tekstiviestin lähetystä varten. Seuraavaksi määritetään parametrin lähetystä varten. Ensiksi käyttäjänimi ja salasana. Näiden jälkeen määritetään smc, eli viestille käytettävä laitteistokokoonpano ja lähettäjän tiedot. Viimeisenä määritetään lähettäjä, vastaanottaja, sekä lähetettävä viesti. Lähettäjän numerolla ei tässä komennossa ole merkitystä, sillä se on määritetty jo Kannelin konfiguraatitiedostossa ja komennossa määritetty numero korvautuu automaattisesti oikealla numerolla. Lähettäjää ei kuitenkaan voi jättää komennosta pois, tai lähetys epäonnistuu.

### 5.3 Hylätyt ratkaisut

Sisäänkirjautumista hallintasivustolle ei toteutettu, koska sivusto tulee todennäköisesti olemaan yrityksen CRM:n https- sivuston alaisuudessa ja pystyy todennäköisesti käyttämään samaa sisäänkirjautumista CRM:n kanssa. Näin ei tarvitse toteuttaa erikseen riittävän turvallista kirjautumistoimintoa.

### 5.4 Testaus

Testiympäristöksi rakennettiin kannettavalle tietokoneelle paikallinen Kannel-serveri. Testauksessa käytettiin Kannel gateway 1.4.3-versiota, joka on viimeisin vakaa versio. Tätä versiota the Kannel Group suosittelee käytettäväksi projekteissa (Kannel julkaisut) ja Ubuntu pohjaista Linux Mint-käyttöjärjestelmää. Tietokantahallintaa varten asennettiin PHPMyAdmin-sovellus, joka mahdollistaa tietokantojen graafisen seurannan ja muokkauksen internetiselaimen kautta. Sms-modeemiksi valittiin Sierra Wireless Air-Link Fastrack Xtend FXT009-malli, hyvistä arvosteluista ja toimintavarmuudesta johtuen. Diafaan.com sivuston mukaan laite sopii erityisen hyvin kriittisiin ja pitkäaikaista toimintavarmuutta vaativiin järjestelmiin, kunhan laitetta käytetään sarjaporttiliitännän kautta. Heidän testiensä mukaan USB-liitäntää käytettäessä toimintavarmuus ei ollut yhtä hyvä. Ulkoinen virtalähteen ja erillisen antenni ansiosta laite sopii myös käytettäväksi palvelinsalissa tai muissa teknisissä tiloissa.

SMS-modeemiin tehtiin testausta varten Integralin matkapuhelinliittymä. Testauksessa suurimmat ongelmat ilmenivät jo testiympäristössä, jo ennen varsinaisia testejä laitteiston ja Kannelin yhteistoiminnassa. Kannelin lähes kaikki tarvittavat määrytykset tehdään smskannel.conf-tiedostossa, joten sen testaukseen ja virheiden korjaukseen menee eniten aikaa.

Kannel käynnistetään suorittamalla seuraavat komennot erillisissä terminaali-ikkunoissa:

1: *gw/bearerbox gw/smskannel.conf*

2: *gw/smsbox gw/smskannel.conf*

Näiden komentojen jälkeen, jos virheitä ei ilmene järjestelmän määrytyksissä, järjestelmä on käynnissä ja testitekstiviestejä voidaan lähettää ja vastaanottaa. Kaikki tapahtumat tulevat näkyviin terminaali-ikkunoihin ja määriteltyihin lokitiedostoihin. Lokitie-

dostoja hallitaan myös kannelin konfigurointitiedostosta. Lokitiedostojen tasoa voidaan vaihtaa välillä 0-5, joista 0 kertoo eniten tapahtumia ja 5 ainoastaan kriittiset virheet. Testauksessa käytettiin lokitasoa 0. Tasolla 0 lokitiedostoista näkyy paljon tietoa tehdyistä toiminnoista. Suuri osa tiedosta on turhaa ja testauksessa tärkeimmät tiedot ovat lähinnä virheet ja varoitukset. Kuvassa 6 Kannelin testaukseen tarvittavat kaksi Linuxin terminaali-ikkunaa auki ja Kannel käynnissä. Kuvassa näkyy vastaanotettu testiviesti. (Kannel käyttöohje, 3)

```

Terminal
File Edit View Search Terminal Help
0,30,"SM",0,30
2014-06-03 19:53:59 [3545] [4] DEBUG: AT2[gas-it]: <-- OK
2014-06-03 19:54:51 [3545] [4] DEBUG: AT2[gas-it]: <-- AT+M
2014-06-03 19:54:51 [3545] [4] DEBUG: AT2[gas-it]: <-- OK
2014-06-03 19:54:51 [3545] [4] DEBUG: AT2[gas-it]: <-- AT+CPMS?^M
2014-06-03 19:54:51 [3545] [4] DEBUG: AT2[gas-it]: <-- +CPMS: "SM",0,30,"SM",
0,30,"SM",0,30
2014-06-03 19:54:51 [3545] [4] DEBUG: AT2[gas-it]: <-- OK
2014-06-03 19:55:43 [3545] [4] DEBUG: AT2[gas-it]: <-- AT+M
2014-06-03 19:55:43 [3545] [4] DEBUG: AT2[gas-it]: <-- OK
2014-06-03 19:55:43 [3545] [4] DEBUG: AT2[gas-it]: <-- AT+CPMS?^M
2014-06-03 19:55:43 [3545] [4] DEBUG: AT2[gas-it]: <-- +CPMS: "SM",0,30,"SM",
0,30,"SM",0,30
2014-06-03 19:55:43 [3545] [4] DEBUG: AT2[gas-it]: <-- OK
2014-06-03 19:56:35 [3545] [4] DEBUG: AT2[gas-it]: <-- AT+M
2014-06-03 19:56:35 [3545] [4] DEBUG: AT2[gas-it]: <-- OK
2014-06-03 19:56:35 [3545] [4] DEBUG: AT2[gas-it]: <-- AT+CPMS?^M
2014-06-03 19:56:35 [3545] [4] DEBUG: AT2[gas-it]: <-- +CPMS: "SM",0,30,"SM",
0,30,"SM",0,30
2014-06-03 19:56:35 [3545] [4] DEBUG: AT2[gas-it]: <-- OK
2014-06-03 19:57:27 [3545] [4] DEBUG: AT2[gas-it]: <-- AT+M
2014-06-03 19:57:27 [3545] [4] DEBUG: AT2[gas-it]: <-- OK
2014-06-03 19:57:27 [3545] [4] DEBUG: AT2[gas-it]: <-- AT+CPMS?^M
2014-06-03 19:57:27 [3545] [4] DEBUG: AT2[gas-it]: <-- +CPMS: "SM",0,30,"SM",
0,30,"SM",0,30
2014-06-03 19:57:27 [3545] [4] DEBUG: AT2[gas-it]: <-- OK
2014-06-03 19:57:54 [3545] [4] DEBUG: AT2[gas-it]: <-- +CMG: ,33
2014-06-03 19:57:54 [3545] [4] DEBUG: AT2[gas-it]: <-- 0701535880770153040C91
534870672529000416030917515210FD470380C7ABB41F4F29C9E06C500
2014-06-03 19:57:54 [3545] [4] DEBUG: AT2[gas-it]: received message from SMSC
: +358508771035
2014-06-03 19:57:54 [3545] [4] DEBUG: AT2[gas-it]: Numeric sender (internatio
nal) <+358407765392>
2014-06-03 19:57:54 [3545] [4] DEBUG: AT2[gas-it]: User data length read as (
15)
2014-06-03 19:57:54 [3545] [4] DEBUG: AT2[gas-it]: Udh decoding done len=15 u
dh=0 udhlen=0 udh=""
2014-06-03 19:57:54 [3545] [7] DEBUG: send_msg: sending msg to box: <127.0.0.
1>
2014-06-03 19:57:54 [3545] [7] DEBUG: boxc_sender: sent message to <127.0.0.1
>
2014-06-03 19:57:54 [3545] [6] DEBUG: boxc_receiver: got ack

Terminal
File Edit View Search Terminal Help
2014-06-03 19:51:24 [3557] [7] DEBUG: Thread 7 (gw/heartbeat.c:heartbeat_thre
ad) maps to pid 3557.
2014-06-03 19:57:54 [3557] [4] INFO: Starting to service <Tama on testi 1> fr
om <+358407765392> to <
>
2014-06-03 19:57:54 [3557] [4] DEBUG: Started thread 8 (glib/fdset.c:poller)
2014-06-03 19:57:54 [3557] [4] DEBUG: Started thread 9 (glib/http.c:write_re
quest_thread)
2014-06-03 19:57:54 [3557] [9] DEBUG: Thread 9 (glib/http.c:write_request_th
read) maps to pid 3557.
2014-06-03 19:57:54 [3557] [9] DEBUG: Queue contains 0 pending requests.
2014-06-03 19:57:54 [3557] [9] DEBUG: Parsing URL 'http://localhost/
sms_receive.php?phone=2B358407765392&text=Tama+on+testi+1':
2014-06-03 19:57:54 [3557] [9] DEBUG: Scheme: http://
2014-06-03 19:57:54 [3557] [9] DEBUG: Host: localhost
2014-06-03 19:57:54 [3557] [9] DEBUG: Port: 80
2014-06-03 19:57:54 [3557] [9] DEBUG: Username: (null)
2014-06-03 19:57:54 [3557] [9] DEBUG: Password: (null)
2014-06-03 19:57:54 [3557] [9] DEBUG: Path: /
/sms_receive.php
2014-06-03 19:57:54 [3557] [9] DEBUG: Query: phone=2B358407765392&text=Tam
a+on+testi+1
2014-06-03 19:57:54 [3557] [9] DEBUG: Fragment: (null)
2014-06-03 19:57:54 [3557] [9] DEBUG: HTTP: Opening connection to 'localhost:
80' (fd=26).
2014-06-03 19:57:54 [3557] [9] DEBUG: Socket connecting
2014-06-03 19:57:54 [3557] [8] DEBUG: Thread 8 (glib/fdset.c:poller) maps to
pid 3557.
2014-06-03 19:57:54 [3557] [8] DEBUG: Get info about connecting socket
2014-06-03 19:57:54 [3557] [8] DEBUG: HTTP: Sending request:
2014-06-03 19:57:54 [3557] [8] DEBUG: Octet string at 0x7fbfb80008e0:
2014-06-03 19:57:54 [3557] [8] DEBUG: len: 150
2014-06-03 19:57:54 [3557] [8] DEBUG: size: 1024
2014-06-03 19:57:54 [3557] [8] DEBUG: immutable: 0
2014-06-03 19:57:54 [3557] [8] DEBUG: data: 47 45 54 20 2f 64 65 78 74 72 6
1 2f 73 6d 73 5f GET /
/sms
2014-06-03 19:57:54 [3557] [8] DEBUG: data: 72 65 63 65 69 76 65 2e 70 68 7
0 3f 70 68 6f 6e receive.php?phon
2014-06-03 19:57:54 [3557] [8] DEBUG: data: 65 3d 25 32 42 33 35 38 34 30 3
7 37 36 35 33 39 e=2B35840776539
2014-06-03 19:57:54 [3557] [8] DEBUG: data: 32 26 74 65 78 74 3d 54 61 6d 6
1 2b 6f 6e 2b 74 2&text=Tama+on+t
2014-06-03 19:57:54 [3557] [8] DEBUG: data: 65 73 74 69 2b 31 20 48 54 54 5
0 2f 31 2e 31 0d esti+1 HTTP/1.1.

```

Kuva 6. Kannel käynnissä, kuvassa vastaanotettu testiviesti

Kuvan terminaali-ikkunoissa on punaisella alleviivattuna olennaisia kohtia. Vasemman puoleisessa terminaali-ikkunassa näkyy tieto vastaanotetusta viestistä ja alempana kerrotaan, että viesti on toimitettu eteenpäin käsiteltäväksi. Oikean puoleisessa ikkunassa on enemmän tietoa viestistä ja viestille tehtävistä toiminnoista. Ylimmäisenä alleviivatuna kohta ”Starting to service <Tama on testi 1>”, jossa kerrotaan toiminnon aloittamisesta viestille ”Tama on testi 1”. Alempana alleviivatulla rivillä kerrotaan viestille suoritettu toiminto: ParsingURL ”http://localhost/sms\_receive.php?

phone%2B358407765392&text=Tama+on+testi+1”; Viesti on lähetetty testiympäristön paikalliselle php-tiedostolle nimeltä sms\_receive.php. Toiminto määriteltiin aiemmin esitellyssä kannelin konfigurointitiedostossa. Merkillä ”?” erotetaan kutsuttu kohdetiedosto ja parametrit, jotka tiedostolle välitetään. Parametrit erotetaan toisistaan merkillä ”&”. Tässä tapauksessa parametreina lähettäjän numero, sekä tekstiviestin koko teksti.

## 5.5 Virhetilanteet

Tässä kappaleessa käsitellään muutamia eteen tulleista virhetilanteista ja niiden ratkaisuja. Suurin osa virheistä johtuu luonnollisesti joko omista kirjoitusvirheistä konfiguraatiotiedostossa, väärästä määritteestä tai laitteistosta puuttuvasta komponentista. Seuraavaksi muutamia esimerkkejä virhetilanteista.

Suoritettaessa käynnistyskomentoa, Kannelin toiselle vaadituista osista omalla smskannel.conf tiedostolla, komennolla: `gw/bearerbox gw/smskannel.conf`

järjestelmä antoi seuraavan virheen:

*2014-04-02 17:04:53 [2758] [6] DEBUG: AT2[Wavecom]: detecting modem speed.*

*2014-04-02 17:04:53 [2758] [6] INFO: AT2[Wavecom]: opening device*

*2014-04-02 17:04:53 [2758] [6] ERROR: AT2[Wavecom]: open failed! ERRNO=2*

*2014-04-02 17:04:53 [2758] [6] ERROR: System error 2: No such file or directory*

Laitteen portti oli määritetty vääräksi: *device = /dev/ttyUSB0*, pitkän selvityksen jälkeen vaihdettu käytettäväksi portiksi: *device = /dev/ttyACM0*. Tämän jälkeen SMS-modeemi tunnistui oikein. Seuraava virhe johtuu Linuxin ominaisuuksista ja tarkasti määritellyistä laitteiden käyttöoikeuksista.

*2014-04-03 03:41:29 [18834] [4] DEBUG: AT2[Wavecom]: detecting modem speed.*

*2014-04-03 03:41:29 [18834] [4] INFO: AT2[Wavecom]: opening device*

*2014-04-03 03:41:29 [18834] [4] ERROR: AT2[Wavecom]: open failed! ERRNO=13*

*2014-04-03 03:41:29 [18834] [4] ERROR: System error 13: Permission denied*

Virhe kertoo selvästi, että ongelma on käyttöoikeuksissa: ”System error 13: Permission denied”. Koska kyse on SMS-modeemin nopeudesta: ”detecting modem speed”, todennäköisesti kyseessä on tämän laitteen käyttöoikeus. Lisätään käyttäjä ryhmään ”dialout”, jolla on oikeus käyttää laitetta. Tämä onnistuu Linuxissa seuraavalla komennolla: `sudo usermod -a -G dialout $USER`

Komennossa `sudo` kertoo, että toiminto tehdään järjestelmänvalvojana. ”Usermod” käsitellään käyttäjäryhmiä. `-a`:lla ilmoitetaan käyttöjärjestelmälle, että halutaan lisätä seuraava käyttäjä tiettyyn ryhmään ja `-G`:llä kerrotaan, ettei valittua käyttäjää poisteta ryhmistä, joihin käyttäjä tällä hetkellä kuuluu.



## 5.6 Käyttöönotto

Käyttöönottoa ei vielä pystytty aikatauluttamaan. Suunnitelmissa on ottaa järjestelmä käyttöön vielä vuoden 2014 aikana. Käyttöönotossa on mahdollisesti vielä omat ongelmansa. Kun hallintasivusto on siirretty yrityksen palvelimelle omaksi sivustoksi tai liitetty yrityksen CRM-järjestelmään osaksi vanhaa kokonaisuutta, lisäksi tarvitaan virtuaalinen Linux-palvelin hallinnoimaan kannelia ja SMS-modeemia. Yrityksen palvelipuolen henkilöiden saatua edellä mainitut kuntoon, tarvitsee enää vanha jo käytössä oleva SIM-kortti siirtää vanhasta puhelimesta uuteen sms-modeemiin. Tämän jälkeen uusi järjestelmä on toimintakunnossa ja päivystysvastuu jakautuu halutuille henkilöille, eivätkä turhat kiireettömät viestit enää häiritse järjestelmän käyttäjiä.

## 6 YHTEENVETO

Olen erittäin tyytyväinen lopputulokseen. Parantamisen varaa löytyy varmasti heti, kun järjestelmä saadaan käyttöön. Tällä hetkellä ja testien perusteella järjestelmä toimii moitteetta, mutta luonnollisesti ongelmia voi ilmaantua, kun järjestelmä siirretään testiympäristöstä sen lopulliselle alustalle Integralin palvelimelle. Sierra Wireless:in SMS-modeemi voi mennä jumiin pitkän käytön aikana. Palvelimella voi tulla levytila täyteen, jolloin kaikki toiminnot pysähtyvät. Tietokannassa, johon tekstiviestit ja vastuuhenkilöiden numerot ja sähköpostiosoitteet tallennetaan, voi ilmetä odottamattomia ja itse järjestelmästä riippumattomia ongelmia. Jos kaikki mahdollisuudet riskit järjestelmässä haluttaisi poistaa, olisi luultavasti toteutettava täysin erillinen varajärjestelmä, joka vian sattuessa kytkeytyisi päälle ja ohittaisi kokonaan tämän varsinaisen järjestelmän.

PHP ohjelmointikielenä on miellyttävä. Ennen tätä työtä olin tutustunut PHP-ohjelmointiin vain vähän, mutta tämän projektin aikana opin pitämään PHP:sta todella paljon. Sisäistäminen on helppoa ja nopeaa ja koodi tehokasta. Tulen varmasti suosimaan PHP:ta myös tulevilla projekteilla.

Työn toteutus oli erittäin antoisa ja opettavainen. Työn jokainen vaihe toi lisää mielenkiintoisia mahdollisuuksia hälytysjärjestelmästä ja parannuksia olisi saanut lisätä jatkuvasti järjestelmämäärittelyyn. Esimerkiksi hallintapaneelin laajennus niin, että kaikki hälytysviestit tulisivat hallintapaneeliin näkyviin. Viestit pystyisi kuittaamaan ensin hoidettavaksi, ettei asiaa hoideta useaan kertaan. Hoidettuaan asian, viesti kuitattaisi vielä valmiiksi, jotta kaikki tietävät asian olevan kunnossa, eikä mitään jää roikkumaan järjestelmään puolivalmiina. Tällä tavalla käyttäjien ei tarvitsisi ilmoitella muille käyttäjille erikseen toimistaan, vaan jokainen voi ennen toimintaa tarkastaa tilanteen hallintapaneelist.

Toinen mielenkiintoinen kehitysidea olisi automaattinen puhelu vastuuhenkilöille. Puhelussa ilmoitettaisiin automaattisesti nauhoitteella hälytyksestä. Puhelu voidaan soittaa puhelurinkiin, johon käyttäjät voivat liittyä ja poistua siitä soittamalla matkapuhelimellaan tiettyyn numeroon. Puheluringin toiminta on myös hälytysjärjestelmää tukeva, sillä ringin puhelimet soivat yhtäaikaan niin kauan, kunnes ensimmäinen vastaa puheluun. Vastannut henkilö olisi vastuussa asian hoidosta. Huonona puolena omatoimiselle rin-

gistä poistumiselle on tietysti mahdollisuus, että kaikki ringin jäsenet poistuvat jostain syystä yhtä aikaa, jolloin hälytys jäisi huomaamatta. Omatoiminen ringistä poistuminen voidaan joko estää, tai sopia että jokainen poistuja soittaa ensin ringiin ja ilmoittaa poistumisestaan. Tapoja on monia ja aina löytyy toimiva ratkaisu. Integral tarjoaa jo vastaavaa järjestelmää asiakkailleen myymissään puhelinjärjestelmissä, joten toteutuskin on täysin mahdollinen.

Lopputulos oli mielestäni hyvä ja toimiva. Se vastasi pitkälti alkuperäistä suunnitelmaa ja toteutti hienosti vaaditut tarpeet.

## LÄHTEET

The Kannel Group. 2009. Sivuston uutiset. Luettu 1.3.2013. <http://kannel.org/news.shtml>

The Kannel Group. 2010. Kannel svn-r5090 User's Guide. Luettu 1.3.2013. <http://www.kannel.org/download/kannel-userguide-snapshot/userguide.html>

Drupal. 2011. Kannel projekti. Luettu 7.2.2014. <https://drupal.org/project/kannel>

W3Schools. PHP esittely. Luettu 5.5.2013. <http://www.w3schools.com/php>

The PHP Group PHP esittely. Luettu 5.5.2013. <http://www.php.net/manual/en/intro-what-is.php>

Integral Oy. 2014. Luettu 15.3.2014. <http://integral.fi/yritys>

Diafaan communication software. SMS-modeemin arvostelu. Luettu 9.8.2013 <http://www.diafaan.com/reviews/sierra-wireless-airlink-fastrack-xtend-gsm-modem>

## ERITYISKIITOKSET!

Mika Puntti  
Tuomas Valtonen  
Jaakko Kallio  
Hannu Salmi